

IN-64
40349
P.15

NASA Technical Memorandum 105203
ICOMP-91-14

(NASA-TM-105203) LEAST-SQUARES SOLUTION OF
INCOMPRESSIBLE NAVIER-STOKES EQUATIONS WITH
THE P-VERSION OF FINITE ELEMENTS (NASA)

15 p

CSCL 12A

N91-31911

Uncles

G3/64 0040349

Least-Squares Solution of Incompressible Navier-Stokes Equations With the P-Version of Finite Elements

Bo-nan Jiang

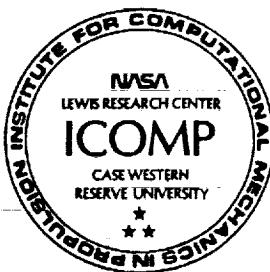
*Institute for Computational Mechanics in Propulsion
Lewis Research Center
Cleveland, Ohio*

and

Vijay Sonnad
*IBM Corporation
Austin, Texas*

September 1991

NASA





LEAST-SQUARES SOLUTION OF INCOMPRESSIBLE NAVIER-STOKES EQUATIONS WITH THE P-VERSION OF FINITE ELEMENTS

Bo-nan Jiang

Institute for Computational Mechanics in Propulsion
Lewis Research Center
Cleveland, Ohio 44135

Vijay Sonnad

IBM Corporation
Austin, Texas 78758

Summary

A p-version of the least-squares finite element method, based on the velocity-pressure-vorticity formulation, is developed for solving steady-state incompressible viscous flow problems. The resulting system of symmetric and positive definite linear equations can be solved satisfactorily with the conjugate gradient method. In conjunction with the use of rapid operator application which avoids the formation of either element or global matrices, it is possible to achieve a highly compact and efficient solution scheme for the incompressible Navier-Stokes equations. Numerical results are presented for two-dimensional flow over a backward-facing step. The effectiveness of simple outflow boundary conditions is also demonstrated.

1. Introduction

In recent years, considerable interest has been shown in the development of the p and the h-p versions of the finite element method[1-3]. The p-version achieves increased accuracy of the approximate solution by increasing the order of the polynomials that make up element shape functions rather than by reducing the mesh size as in the h-version which uses relatively low order polynomials within each element. If the increase in polynomial order and the reduction in element size is made simultaneously or selectively according to local error criteria, we obtain the h-p version. The h-p version can achieve a high level of accuracy with relatively few degrees of freedom.

Although the h-version is very flexible in its ability to handle complex geometries, h-adaptive calculations are difficult to implement, since the data structure to handle mesh refinements is not trivial. The p-version of finite elements was designed in order to make adaptive computations easier than with the h-version. While it is possible to achieve excellent results with higher-order Lagrange type shape functions[4], there are significant simplifications in programming effort when Legendre type shape functions are used. A particular choice of functions uses linear combinations of Legendre polynomials leading to hierachic shape functions. This yields relatively well conditioned problems and nested matrices[5]. If the underlying solution is smooth, the convergence of the p-version to the solution is rapid, so that it is possible to reach a given level of accuracy with lesser degrees of freedom than with the h-version. Because of these advantages, the p-version and the h-p version are becoming increasingly popular in a variety of applications, especially for problems in solid mechanics.

The finite element method has been employed in the solution of problems in incompressible fluid dynamics for almost two decades. In principle, any finite element method for incompressible viscous flows, (a review of some recent developments in this area are contained in a survey paper by Gresho[6], and in a monograph by Gunzberger[7]), can be realized by the p-version. However, in practice, it is not easy to employ standard approaches with the p-version of the finite element method. Commonly used finite element methods discretize the equations of incompressible fluid flow by the classical Galerkin mixed method, leading to a system of algebraic equations. The matrix corresponding to this system of equations is nonsymmetric due to the convection terms in the momentum equations, and non-positive-definite owing to the uncoupled nature of the continuity equation. The solution of this class of equations by existing iterative methods is not satisfactorily robust for many problems of practical interest, and direct factorization is considered the only viable approach for solving such systems of equations. But for large 2D and 3D problems, the computer storage required by a direct method becomes prohibitively large. For the p-version, this difficulty becomes more severe, since the element matrices of the p-version method are large and dense.

For solving steady-state incompressible viscous flow problems, an alternate approach

is the least-squares method, based on minimization of the residuals of the first-order equations written in the velocity-pressure-vorticity formulation[8-10]. This method can accommodate equal-order elements. For the linear Stoke's equations it has been shown that this method has an optimal rate of convergence for velocity, pressure and vorticity[11]. This method has the important advantage of yielding discrete equations that are symmetric and positive definite and can be solved effectively by iterative methods. The use of iterative methods, such as the conjugate gradient method, can reduce significantly the storage requirement and computing time, because they work by improving an initial guess, and do not require the formulation and factorization of a global matrix. Therefore, in this study we incorporate the least-squares method with the p-version finite elements. The paper is organized as follows. In Section 2 we describe briefly the least-squares method. In Section 3 we explain the idea of rapid operator application which allows significant reductions in the computation time and storage requirements for the solution of large problems with the p-version of the finite elements. In Section 4, we present results for the 2D flow over a backward-facing step using simple outflow boundary condition. In the final section we end with some concluding remarks.

2. The Least-Squares Method for the Incompressible Navier-Stokes Equations

The 2D steady-state incompressible Navier-Stokes equations can be written in the velocity-pressure formulation as:

$$\begin{aligned} \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0, \\ u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} - \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) &= 0 \quad \text{in } \Omega, \\ u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} - \frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) &= 0. \end{aligned} \quad (1)$$

Here all variables are nondimensionalized; (u, v) are the velocity components, p is the pressure, and Re denotes the Reynolds number. Appropriate boundary conditions should be supplied to ensure that this is a well-posed boundary value problem.

The classic Galerkin approach to discretizing these equations leads to nonsymmetric, nonpositive definite equations. One way to avoid this difficulty is to use the least-squares method upon this system of equations.

Let us examine the general least-squares approach. Consider a boundary value problem:

$$\mathbf{L}\mathbf{u} = \mathbf{f} \quad \text{in } \Omega,$$

$$(2) \quad \mathbf{B}\underline{u} = \underline{g} \quad \text{on} \quad \Gamma,$$

where \mathbf{L} is a general partial differential operator, \mathbf{B} is a boundary operator, \underline{u} is an unknown vector, \underline{f} is a given vector-valued function, and \underline{g} is a given vector-valued function on the boundary.

For an arbitrary trial function \underline{u} which satisfies the boundary condition $\mathbf{B}\underline{u} = \underline{g}$ on Γ , we define the residual function: $\underline{R} = \mathbf{L}\underline{u} - \underline{f}$. The least-squares method minimizes the following functional:

$$I(\underline{u}) = \int_{\Omega} \underline{R}^T \underline{R} d\Omega = \int_{\Omega} (\mathbf{L}\underline{u} - \underline{f})^T (\mathbf{L}\underline{u} - \underline{f}) d\Omega. \quad (3)$$

Taking the variation of I with respect to \underline{u} and setting $\delta I = 0$ and $\delta \underline{u} = \underline{w}$, leads to the least-squares weak statement: find \underline{u} satisfying $\mathbf{B}\underline{u} = \underline{g}$ on Γ such that :

$$\int_{\Omega} (\mathbf{L}\underline{w})^T (\mathbf{L}\underline{u} - \underline{f}) d\Omega = 0. \quad (4)$$

The least-squares weak statement corresponds to the Galerkin statement with the test functions \underline{w} being replaced by $\mathbf{L}\underline{w}$.

The finite element discretization of the least-squares weak statement(4) yields symmetric, positive definite equations; however the condition number is of the order of the square of that obtained by Galerkin discretization, which can lead to slow convergence, or lack of convergence. Furthermore, in this case the operator \mathbf{L} contains second-order derivatives, it is then necessary to use C^1 continuous shape functions, which are very awkward to use.

To avoid both of the above problems the least-squares method is applied to the Navier-Stokes equations recast as first-order partial differential equations:

$$\begin{aligned} \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0, \\ u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + \frac{\partial p}{\partial x} + \frac{1}{Re} \frac{\partial \omega}{\partial y} &= 0, \\ u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + \frac{\partial p}{\partial y} - \frac{1}{Re} \frac{\partial \omega}{\partial x} &= 0 \quad \text{in} \quad \Omega, \\ \omega + \frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} &= 0. \end{aligned} \quad (5)$$

In the above equations, the vorticity ω is introduced purely in order to express the Navier-Stokes equations as a system of first-order partial differential equations while introducing the minimum number of additional variables. Since the system(5) is of first-order, the boundary conditions are very simple, and do not involve the derivatives of unknowns. A crucially important point is that *even though vorticity is introduced as a variable, it is not required to specify the vorticity on a boundary for the first-order equations(5), if the boundary conditions for the original set of equations(1) are related to only the velocity components (u, v) and the pressure p , and not to their derivatives.*

While this approach may be used with the h-version of finite elements, or even the finite difference method, we combine the least-squares method with the p-version of finite elements. In the next section we discuss the implementation of the method.

3. Iterative Solution with Rapid Operator Application

We will show in this section that the use of well known iterative methods in conjunction with rapid operation techniques, leads to a solution scheme that can achieve significant savings in storage for the solution of problems with the p-version. We consider here the conjugate gradient approach which can be applied to our case, because the matrices are symmetric and positive definite.

The major computational component in this method is the multiplication of the global matrix by a global vector. Since the global matrix is formed by assembling individual element matrices, the multiplication can be done without forming the global matrix. This alone can greatly reduce the storage requirements over direct methods. This is one of the components of the element-by-element approach[12].

If the storage is to be minimal, the element matrices should not be stored but should be recomputed at every iteration. With the h-version this is not very expensive, and it is possible to reform the element matrices at every iteration without paying a high penalty. However, with the p-version, the reformulation of the element matrix at every iteration, (particularly for 3-D problems with high levels of polynomials), can be exorbitantly expensive in solution time. Furthermore, even a single element matrix can take up a large amount of storage; e.g., for a 3-D structural analysis problem using polynomials of order 9, a single element matrix occupies about 36 Mbytes when the symmetric part of the matrix is stored in double precision.

We now show that with the use of simple and well known techniques, it is possible to get the matrix vector product without forming and storing the element matrices thus making the storage requirements close to minimal. The rapid operator application (in effect, the matrix vector product) is performed with *less operations than when an element matrix is formed, i.e., there is no penalty in computing time.*

The key ingredient is the use of elements that employ tensor-product shape functions. Since such shape functions implicitly perform separation of variables, it is possible to group terms so that several terms are reused in the rapid operator application. The savings associated with the use of tensor-product shape functions are best illustrated with a simple example. Consider a rectangular element within which we define b^2 shape functions. The variation of a quantity, e.g., u , within this element can be expressed in the form

$$u = U_\alpha \Psi_\alpha(x, y) \quad (\alpha = 1, b^2), \quad (6)$$

where U_α are the coefficients and $\Psi_\alpha(x, y)$ are the shape functions. (In the above equation, repeated indices imply summation.) When the shape functions are in tensor-product form, each two-dimensional shape function is expressed as a product of one-dimensional shape functions. Thus, the above expression can be rewritten as

$$u = \phi_l(x) \phi_m(y) U_{lm} \quad (l, m = 1, b). \quad (7)$$

To illustrate the evaluation of an operator with these shape functions, we consider the evaluation of the product of the mass matrix by a vector; i.e., we wish to evaluate for values of $\alpha = 1, b^2$, the expression

$$D_\alpha = M_{\alpha\beta} U_\beta \quad (\beta = 1, b^2), \quad (8)$$

where the mass matrix is given by

$$M_{\alpha\beta} = \int_{-1}^1 \int_{-1}^1 \Psi_\alpha(x, y) \Psi_\beta(x, y) dx dy. \quad (9)$$

We need for $\alpha = 1, b^2$

$$D_\alpha = \int_{-1}^1 \int_{-1}^1 \Psi_\alpha(x, y) \Psi_\beta(x, y) U_\beta dx dy \quad (b = 1, b^2). \quad (10)$$

Using the tensor-product form, we expand the above expression to obtain

$$D_{rs} = \int_{-1}^1 \int_{-1}^1 \phi_r(x) \phi_s(y) \phi_l(x) \phi_m(y) U_{lm} dx dy \quad (l, m = 1, b), \quad (11)$$

where r, s each assumes all values between 1 and b . When we perform numerical integration, we replace integrals by weighted sums of the functions evaluated at the integration points. For simplicity, we assume that the number of integration points is equal to the number of shape functions. (This is not necessary, but greatly simplifies the explanation.) Thus, the above expression reduces to

$$D_{rs} = \phi_r(x^i) \phi_s(y^j) \phi_l(x^i) \phi_m(y^j) w(i) w(j) U_{lm} \quad (i, j, l, m = 1, b), \quad (12)$$

where x^i, y^j are the integration points and $w(i), w(j)$ are the weights. We note here that quantities such as $\phi_r(x^i)$ are merely the one dimensional shape functions evaluated at the integration points and are matrices which we denote by ϕ_{ri} . We therefore need to evaluate for all values of r, s from 1 to b , the expression

$$D_{rs} = \phi_{ri} \phi_{sj} w(i) w(j) \phi_{li} \phi_{mj} U_{lm} \quad (i, j, l, m = 1, b). \quad (13)$$

This is done by contracting the indices from the right, one step at a time. Thus, the operation count in evaluating the product of the mass matrix and a vector is now reduced to order (b^3) . By comparison, if the mass matrix were to be formed, it would have required order (b^4) storage. The multiplication by a vector of length b^2 , would take b^4 operations.

The key point is that to evaluate the matrix-vector products, the element matrices are never formed, but rather computed *on the fly*, with numerical quadrature implicit in the evaluation. The savings associated with tensor-product shape functions are well known in mathematics (the multidimensional Fourier expansions is a good example), and in computer aided design[13,14]. In recent years such shape functions have been elegantly exploited by Karniadakis, Orszag and his associates in the spectral element method used for fluid flow computations [15]; similar results have been obtained in solid mechanics[16].

The above derivation does not take into account the specific nature of the shape functions other than that they are in tensor product form. The shape functions for the p-version are chosen so that their derivatives are nearly orthogonal over a cube, and the evaluation of all D_{rs} could be performed without numerical integration in order b^2 operations. However, the intention above is to illustrate the approach that is taken with more general curved elements, where a Jacobian is present within the integral so that orthogonality is lost.

The approach described above can be used to evaluate the operator corresponding to the 2-D incompressible Navier-Stokes equations. The unknowns are the two components of the velocity u, v , the vorticity ω , and the pressure p . If we consider a single element with b^2 shape functions, the size of an element matrix is approximately $4b^2 * 4b^2$. Multiplication by a vector of length $4b^2$ would take $16b^4$ operations, if advantage is not taken of the tensor product nature of the shape functions. When we use rapid operator application with the number of Gauss points equal to the number of shape functions, it can be shown that the number of floating point operations required are approximately

$$32 * b^3 + \text{Order}(b^2) \text{terms}$$

This estimate still holds when the number of integration points is different from the number of shape functions, except that the constant is no longer 32. The value 32 is also related to the number of terms in the partial differential equation, and the ways in which they can be grouped together; a simpler equation with a lesser number of terms would have a lower constant. This approach is valid for elements that are distorted and also for nonlinear problems. We note here that when required, the element matrix can be formed in order b^5 operations, as compared to b^6 operations with the standard approach.

The program KGNFEM is an experimental finite element code that incorporates these ideas and serves as a framework for both fluid flow and structural analysis computations. We present below some results showing that large scale fluid flow computations can be performed with very modest storage requirements.

4. Problem Description and Results

The approach described above has been tested by solving the problem of flow past a backward-facing step. The back-facing step example has been chosen so as to be able to compare the results with the benchmark solution of Gartling[17]. The step has a height of 0.5 and the channel downstream of this step has a height of 1.0. The total length L of the channel is 30.

The boundary conditions are as follows: At the inlet, a parabolic velocity profile is specified for u , with the tangential velocity v set to zero. At the exit the only outflow boundary condition is $p = 0$. Along the walls, the no slip condition is used. Note that *there are no boundary conditions specified for vorticity*.

The Reynolds number $Re = u_{avg}H/\nu$ was 800 based on the height $H = 1.0$ of the channel, the average inlet velocity $u_{avg} = 0.6667$, and the adjusted viscosity ν .

A total of 20 uniform finite elements were used in the solution of this problem. The orders of the polynomials used for the variables u, v, w, p were 15, 15, 14, and 14 respectively, resulting in a total of 14607 unknowns requiring about 1.25 Mbytes to solve the problem. (The equal orders 15, 15, 15, 15 were also tested; we didn't find any visible difference between these two results.) Roughly speaking, the required storage is about 11 times of number of unknowns in double precision words.

In this numerical experiment, a simple successive substitution scheme is used to obtain the solution. At the beginning we assume that $u = v = 0$. The required number of substitutions is 80 for the L_2 norm of the residual to be less than the tolerance 10^{-4} . In each substitution step, the linear algebraic system is solved by the preconditioned conjugate gradient method. Here we use the simple Jacobi(diagonal) preconditioner.

The computed results (velocity contours, pressure contours, and vorticity contours) are shown in Figure 1. We note that there are no oscillations in the computed results. We also should mention that we didn't use any postprocessing to smooth the computed results. The re-attachment length of the circulating zone immediately behind the step and the detachment and re-attachment points on the upper wall of the channel compare well with the Gartling's results[17].

In order to test the effectiveness of the outflow boundary condition, we cut the length of the backward-facing step problem to a half. We imposed the condition $p = 0$ on the exit boundary. Although $p = \text{constant}$ is not compatible with the real physics at that location,

the method still gives reasonably good solutions and captured the two recirculation zones (see Figure 2).

These numerical results verify the validity of the least-squares method of the Navier-Stokes equations using the p-version of the finite elements.

5. Conclusions

The Navier-Stokes equations have traditionally been the prime generators of nonsymmetric systems of equations. The least-squares approach described in this paper shows that it is possible to generate symmetric, positive definite systems of equations which can be robustly solved by the conjugate gradient method. The least-squares method does not require the use of artificial viscosity or upwind differencing. This method allows extremely simple boundary conditions; It is a clean, robust and consistent approach to solve the Navier-Stokes equations in primitive variable form. In conjunction with the the p-version of finite elements, and the rapid operator application, this approach leads to minimal storage requirements even for large-scale problems.

Acknowledgement

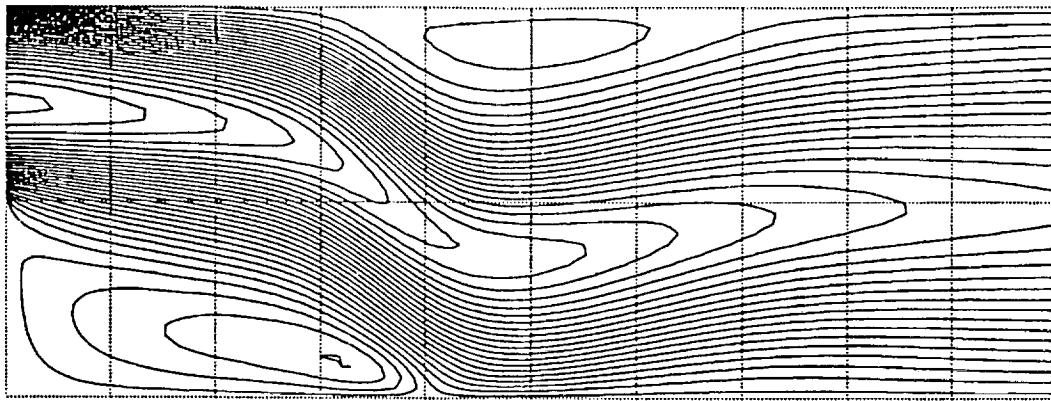
The authors are grateful to Dr. Erico Clementi for providing support and encouragement for this work. This work was performed within the Department of Scientific Computing at IBM Kingston, New York.

References

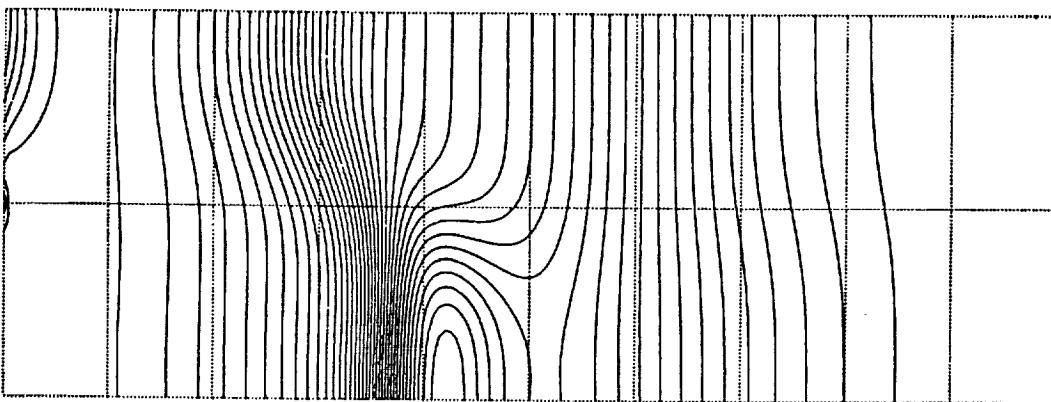
1. I.Babuška, B.A.Szabo, and I.N.Katz, The p-version of the finite element method, *SIAM Journal of Numerical Analysis* 18 (1981) 512-545.
2. I.Babuška and M.Suri, The p- and h-p versions of the finite element method, An overview, *Comput.Meth.Appl.Mech.Engng.* 80 (1990) 5-26.
3. J.T.Oden and L.Demkowicz, Advances in adaptive improvements: a survey of adaptive methods in computational fluid dynamics, *State of the Art Surveys in Computational Mechanics*, Edited by A.K.Noor and J.T.Oden, American Society of Mechanical Engineering, N.Y., 1988.
4. K.Kato and K.J.Bathe, On the use of higher order finite elements in structural mechanics", Report 89-4, Finite Element Research Group, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, Oc-

tober 1989.

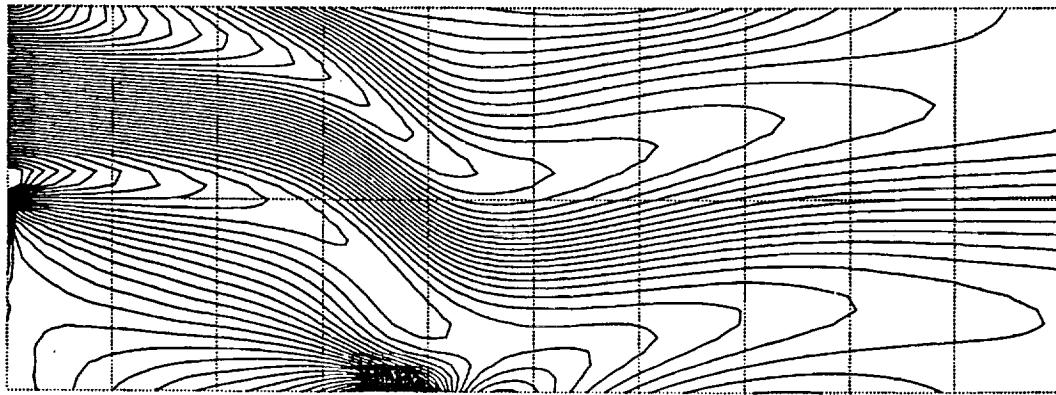
5. B.A.Szabo, and I.Babuška, *Finite Element Analysis*, John Wiley and Sons, New York, 1991.
6. P.M.Gresho, Some current CFD issues relevant to the incompressible Navier-Stokes equations, *Comput.Meth.Appl.Mech.Engng.* (in press).
7. M.D.Gunzberger, *Finite Element Methods for Viscous Incompressible Flows*, Academic Press, 1989.
8. B.N.Jiang and C.L.Chang, Least-squares finite elements for Stokes problem, *Comput.Meth.Appl.Mech.Engng.* 78 (1990) 297-311.
9. B.N.Jiang and L.A.Povinelli, Least-squares finite element method for fluid dynamics, *Comput.Meth.Appl.Mech.Engng.* 81 (1990) 13-37.
10. B.N.Jiang, A least-squares finite element method for incompressible Navier-Stokes problems, *Inter.J.Numer.Meth. Fluids* (in press).
11. C.L.Chang and B.N.Jiang, An Error analysis of least-squares finite element method of velocity-pressure-vorticity formulation for Stokes problem, *Comput.Meth.Appl.Mech. Engng.* 84 (1990) 247-255.
12. G.F.Carey and B.N.Jiang, Element-by-element linear and nonlinear solution schemes, *Commun.Appl.Numer. Methods* 2 (1986) 145-153.
13. S.A.Coons, Surfaces for computer aided design of space forms, M.I.T., Project MAC, MAC-TR-41, June 1967.
14. P.E.Bezier, Example of an existing system in the motor industry: the unisurf system, *Proceedings of the Royal Society A321* (1971) 207-218.
15. G.E.Karniadakis and S.A.Orszag, Spectral simulations of complex flows, *Science and Engineering on Supercomputers*, Edited by E.J.Pitcher, Computational Mechanics Publications, Springer-Verlag, Berlin, 1990.
16. V.Sonnad, S.Foresti, S.Hassanzadeh, and H.Murakami, Minimal storage solution of large 3-D problems using rapid operator application with the p-version of finite elements, IBM Kingston Research Report, KGN-209, March 1990.
17. D.K.Gartling, A test problem for outflow boundary conditions-flow over a backward-facing step, *Int.J.Numer.Methods Fluids* 11 (1990) 951-952.



(a) Velocity

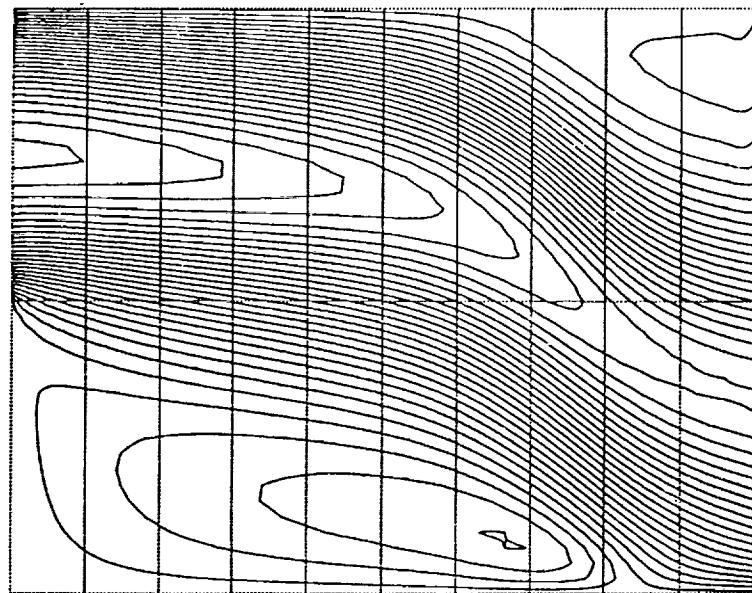


(b) Pressure

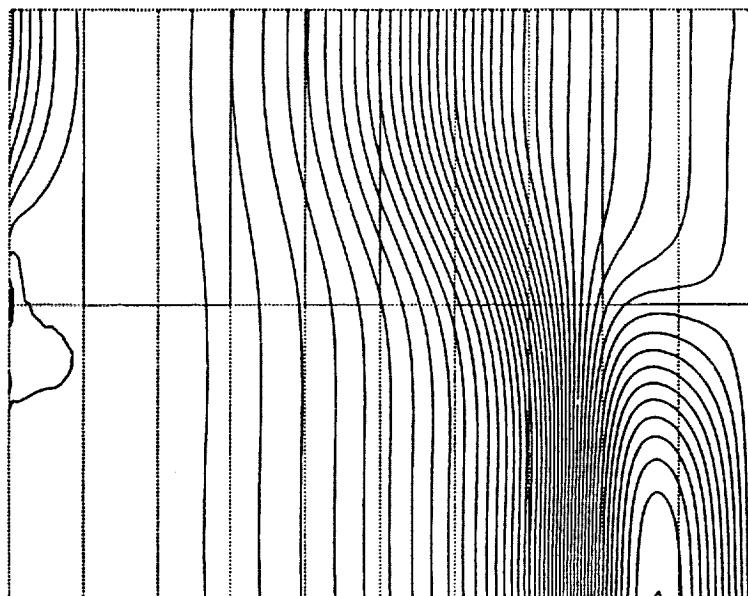


(c) Vorticity

Fig. 1. Computed results for backward-facing step with $L = 30$ at $Re = 800$.

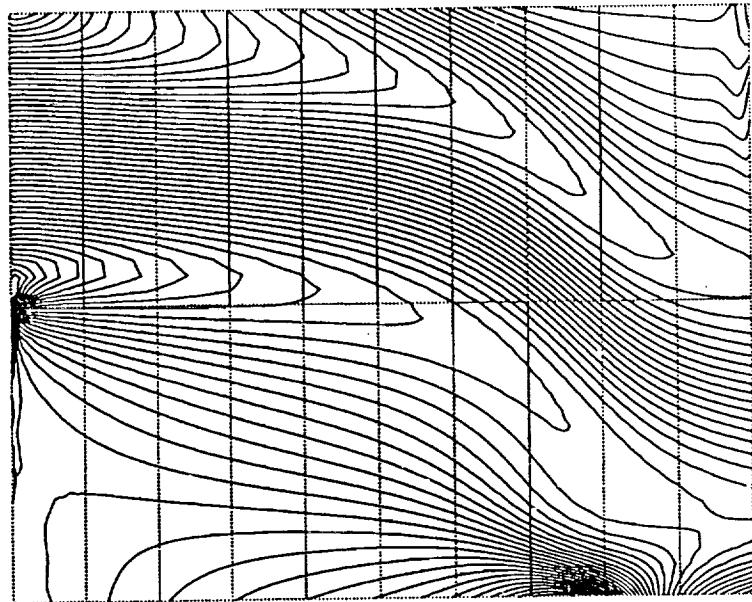


(a) Velocity



(b) Pressure

Fig. 2. Computed results for backward-facing step with $L = 15$ at $Re = 800$ (continued).



(c) Vorticity

Fig. 2. Computed results for backward-facing step with $L = 15$ at $Re = 800$.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</p>			
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED	
	September 1991	Technical Memorandum	
4. TITLE AND SUBTITLE		5. FUNDING NUMBERS	
Least-Squares Solution of Incompressible Navier-Stokes Equations With the P-Version of Finite Elements		WU - 505 - 62 - 21	
6. AUTHOR(S)		8. PERFORMING ORGANIZATION REPORT NUMBER	
Bo-nan Jiang and Vijay Sonnad		E - 6506	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)		10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135 - 3191		NASA TM - 105203 ICOMP 91 - 14	
9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)		11. SUPPLEMENTARY NOTES	
National Aeronautics and Space Administration Washington, D.C. 20546 - 0001		Bo-nan Jiang, Institute for Computational Mechanics in Propulsion, Lewis Research Center (work funded by Space Act Agreement C - 99066 - G). Vijay Sonnad, IBM Corporation, Austin, Texas 78758. Space Act Monitor: Louis A. Povinelli, (216) 433 - 5818.	
12a. DISTRIBUTION/AVAILABILITY STATEMENT		12b. DISTRIBUTION CODE	
Unclassified - Unlimited Subject Category 64			
13. ABSTRACT (Maximum 200 words)			
<p>A p-version of the least-squares finite element method, based on the velocity-pressure-vorticity formulation, is developed for solving steady-state incompressible viscous flow problems. The resulting system of symmetric and positive definite linear equations can be solved satisfactorily with the conjugate gradient method. In conjunction with the use of rapid operator application which avoids the formation of either element or global matrices, it is possible to achieve a highly compact and efficient solution scheme for the incompressible Navier-Stokes equations. Numerical results are presented for two-dimensional flow over a backward-facing step. The effectiveness of simple outflow boundary conditions is also demonstrated.</p>			
14. SUBJECT TERMS		15. NUMBER OF PAGES	
Least squares; Finite element; Incompressible; Navier-Stokes; Viscous flow; Velocity-pressure-vorticity; P-version		14	
		16. PRICE CODE	
		A03	
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	